```
1
2
3   from a0_items import *
4
5
6
7   # Write list of url
8   def write_url (code_file_path, url_str, url_count):
9
10     os.remove (code_file_path) #remove file before write
11
12     with open (code_file_path, 'a+') as f:
13
14        for x in range (request_count):
15
16           f.write(url_str + '\n')
17  '''
18  url = "https://hello_world"
19  write_url (code_file_path='test_code.py', url_str=url, url_count=5)
20  '''
21
22
23  # Write list of function names
24  def write_func_names (code_file_path, func_name, func_name_count):
25
26     os.remove (code_file_path) #remove file before write
27
28     with open (code_file_path, 'a+') as f:
29
30        for x in range (func_name_count):
31
32           f.write(func_name + '_' + str(x) + ' ( )' + '\n')
33
34  '''
35  func_str = 'test_func'
36  write_func_names (code_file_path='test_code.py', func_name=func_str,
    func_name_count=5)
37  '''
38
39
40  ##############################################################################
    ##
41  ##### Write url requests - see plg_Request_url
42  ##############################################################################
    ##
43
44
45
46
47
48
49
50
51
52
53
54  ##############################################################################
    ########
55  ### Write multi-processing
56  ##############################################################################
    ##########
57
58  def write_multi_proc_func (code_file_path, M_func_name, func_name, M_func_count):
59
60      os.remove (code_file_path) #remove file before write
61
62      with open (code_file_path, 'a+') as f:
```

```
63
64          f.write('\n \n \n \n')
65          f.write(M_func_name + ' ():')
66          f.write('\n \n')
67
68          for x in range (M_func_count):
69
70            f.write('   p' + str(x) + ' = ' + 'Process(' + func_name + '_' + str(x) + ')')
71            f.write('\n')
72            f.write('   p' + str(x) + '.start()')
73            f.write('\n \n')
74
75  #write_multi_proc_func (code_file_path='test_code.py', M_func_name='def Multiproc',
    func_name='shane_func', M_func_count=10)
76
77
78
79  ##########################################################################################
    ####
80  #### Write concurrent futures
81
82  # https://www.youtube.com/watch?v=IEEhzQoKtQU
83  #
    https://www.packetswitch.co.uk/what-is-concurrent-futures-and-how-can-it-boost-your-pytho
    n-performance/
84  # https://superfastpython.com/processpoolexecutor-common-errors/
85
86  ##########################################################################################
    ###
87
88
89  # Write a list of concurrent future functions
90  def write_con_futures_funcs (code_file_path, C_func_name, func_name, list_name,
    C_func_count):
91
92     os.remove (code_file_path) #remove file before write
93
94     with open (code_file_path, 'a+') as f:
95
96        f.write('\n \n \n \n')
97
98        for x in range (C_func_count):
99
100          f.write(C_func_name + '_' + str(x) + ' ():')
101          f.write('\n \n')
102
103          f.write('   with concurrent.futures.ProcessPoolExecutor() as executor:')
104          f.write('\n \n')
105
106          f.write('      executor.map(' + func_name + '_' + str(x) + ',' + list_name + '_'
              + str(x) + ')')
107          f.write('\n \n')
108
109          f.write('   t2 = time.perf_counter()')
110          f.write('\n')
111
112          f.write("   print" + "(f' finished in {t2} seconds')")
113          f.write('\n \n \n')
114
115
116 #write_con_futures_funcs (code_file_path='test_code.py', C_func_name='def concur_func',
    func_name='shane_func', list_name='LIST_email', C_func_count=10)
117
118
119
120 #Run concurrent future function one by one
121 def write_con_futures_run_SEQUENCE (code_file_path, C_func_name, func_name, C_func_count
```

```
    ):
122
123     os.remove (code_file_path) #remove file before write
124
125     with open (code_file_path, 'a+') as f:
126
127         f.write('\n \n \n \n')
128         f.write(C_func_name + ' ():')
129         f.write('\n \n')
130
131         for x in range (C_func_count):
132
133             f.write('    ' + func_name + '_' + str(x) + ' ()')
134             f.write('\n')
135
136 #write_con_futures_run_SEQUENCE (code_file_path='test_code.py', C_func_name='def
    Con_func', func_name='shane_func', C_func_count=10 )
137
138
139
140 # Run all concurrent future functions concurrently
141 def write_con_futures_run_ALL (code_file_path, C_func_name, func_name, C_func_count ):
142
143     os.remove (code_file_path) #remove file before write
144
145     with open (code_file_path, 'a+') as f:
146
147         f.write('\n \n \n \n')
148         f.write(C_func_name + ' ():')
149         f.write('\n \n')
150         f.write('    with ThreadPoolExecutor(' + str(C_func_count) + ') as ex:')
151         f.write('\n')
152         f.write('        futures = []')
153         f.write('\n')
154
155         for x in range (C_func_count):
156
157             f.write('        futures.append(ex.submit(' + func_name + '_' + str(x) + '))')
158             f.write('\n')
159
160 #write_con_futures_run_ALL (code_file_path='test_code.py', C_func_name='def Con_func',
    func_name='shane_func', C_func_count=10 )
```